

Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Zadání bakalářské práce

Student: **Tomáš Widlák**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: MonkeyData, s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Mgr. Jiří Dvorský, Ph.D.**

Konzultant bakalářské práce: Ing. Rostislav Kresinger

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015



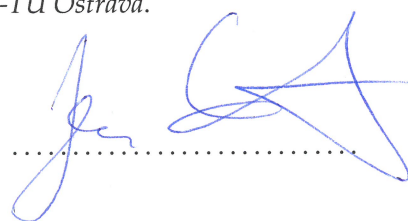
doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava*.

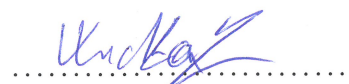
V Ostravě 4. května 2015



.....

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 4. května 2015



.....

Rád bych poděkoval společnosti MonkeyData s.r.o. za to, že mi umožnila u nich vykonat odbornou praxi. Za vřelé začlenění do pracovního týmu i do toho společenského.

Především bych chtěl poděkovat členům vývojového týmu Ing. Rostislavu Kreisingerovi a Tomáši Staňkovi za spoustu cenných rad a také za to, že se se mnou podělili o jejich dlouholeté zkušenosti s vývojem softwaru.

Abstrakt

Tato bakalářská práce popisuje činnost, vykonanou během odborné stáže ve společnosti MonkeyData, s.r.o, která se zabývá vývojem webové aplikace pro analýzu dat z podnikání. Hlavní náplní praxe bylo vyvinout část aplikace, která slouží k efektivnímu získání dat z úložiště a zobrazit je do grafů ve webové aplikaci. Součástí tohoto úkolu bylo potřeba navrhnout vhodnou databázovou strukturu, serverovou část a také klientskou část. Během vývoje části aplikace, která mi byla zadána v rámci odborné praxe, jsem spolupracoval i na ostatních částech aplikace jako je import dat, API a další.

Klíčová slova: MonkeyData, s.r.o, bakalářská praxe, PHP, framework Laravel, MySQL, JavaScript, JQuery, HTML, CSS , LESS, webová aplikace, graf, analytika

Abstract

This thesis describes the work performed during internships at MonkeyData s.r.o, a developer of web application for analysis of data from business. The main scope of practice has been to develop a part of the application that is used to efficiently obtain data from the repository and display them in graphs in a web application. A part of this task was necessary to design an appropriate database structure, the server part and the client part. During a development of a part of an application that was given to me, I also worked on other parts of the application, such as importing data, API and others.

Keywords: MonkeyData, s.r.o, bachelor practice, PHP, framework Laravel, MySQL, JavaScript, JQuery, HTML, CSS , LESS, web application, graph, analytics

Seznam použitých zkratk a symbolů

AJAX	– Asynchronous JavaScript and XML
API	– Application Programming Interface
CSS	– Cascading Style Sheets
dxChart	– JavaScript library
GET	– HyperText Markup Language
HTML	– HyperText Markup Language
HTTP	– Hypertext Transfer Protocol
HTTPS	– Hypertext Transfer Protocol Secure
jQuery	– JavaScript library
JS	– JavaScript
JSON	– JavaScript Object Notation
Laravel	– PHP framework
LESS	– Dynamic stylesheet language
MySQL	– Databaze založená na dialektu SQL
OOP	– Object-oriented programming
PHP	– Hypertext Preprocessor
REST	– Representational State Transfer
SQL	– Structured Query Language
URL	– Uniform Resource Locator
VŠB	– Vysoká škola báňská - Technická univerzita Ostrava
XML	– Extensible Markup Language

Obsah

Úvod	3
1 Firma MonkeyData, s.r.o	4
2 Pracovní zařazení	5
2.1 Firemní prostředí	5
3 Technologie	6
3.1 Databázová část	6
3.2 Serverová část	6
3.3 Klientská část	6
3.4 Architektura	7
4 Projekt MonkeyData	8
4.1 Popis projektu	8
4.2 Členění projektu	8
5 Zvolené řešení	10
5.1 Databázová část	10
5.2 Serverová část	10
5.3 Klientská část	12
5.4 Časový plán	16
6 Využité a nedostatečné znalosti	17
7 Závěr	18
Literatura	19

Seznam obrázků

1.1	Logo MonkeyData s.r.o.	4
5.1	Typ grafu Pie	13
5.2	Typ grafu Spline	14
5.3	Typ grafu Bar	14
5.4	Typ grafu List	15
5.5	Typ grafu Star	15

Úvod

Tématem této práce je seznámit čtenáře s náplní mé odborné praxe vykonané během bakalářského studia ve firmě MonkeyData, s.r.o.

V první části práce seznamuji s firmou MonkeyData. Dále je zde popsáno pracovní zařazení a firemní prostředí, ve kterém se samotná praxe odehrávala. V kapitole 3 je výčet nejdůležitějších technologií, se kterými jsem se setkal při řešení úkolu, které mi v rámci praxe byly zadány.

Kapitola 4 popisuje aplikaci, do jejíž vývoje jsem byl začleněn. Je zde popsáno k čemu samotná aplikace slouží a k čemu se v praxi využívá. Také je zde popsáno základní technologické členění aplikace.

V poslední části práce jsem zhodnotit celý průběh odborné praxe. Zesumarizovat zkušenosti a znalosti získané během praxe a porovnat je ze zkušenostmi nabytými během bakalářského studia na vysoké škole.

1 Firma MonkeyData, s.r.o

Společnost MonkeyData s.r.o. se zabývá především vývojem a provozem aplikace MonkeyData, která má za cíl analýzu dat z online podnikání a prezentaci dat uživatelům ve srozumitelné a pochopitelné podobě pomocí přehledných grafů. Dále se snaží seznámit veřejnost s výhodami analyzování podnikání, např. pomocí online blogu apod.



Obrázek 1.1: Logo MonkeyData s.r.o.

Vedení, vývojářský tým i marketing sídlí v kancelářích v rámci komplexu BIC Ostrava s.r.o.

Zástupci společnosti se zúčastnili mnoha tuzemských, ale i zahraničních konferencí zaměřené na startup-ové firmy, či konferencí zaměřujících se na online podnikání, analýzu dat nebo konference spojené s internetovým prostředím jako takovým.

2 Pracovní zařazení

Při nástupu na odbornou praxi jsem byl zařazen do vývojového týmu na pozici PHP developer, posléze jsem byl využit jako Frontend a JavaScript Developer. Z počátku praxe mi byl velice nápomocen můj mentor Ing. Rostislav Kreisinger a také kolega Tomáš Staňek.

2.1 Firemní prostředí

Byl jsem začleněn do tříčlenného vývojového týmu. Při nástupu mi byl projektovým managerem vytvořen firemní email, na základě kterého jsem získal přístup do všech systémů, které firma používá ke komunikaci, firemním procesům a pro přístup k administraci cloudu, ve kterém běží celá aplikace.

K interní komunikaci se využívá několik komunikačních prostředků. První a ten nejdůležitější, jak mi to bylo prezentováno, je osobní kontakt. Proto jsem se ho snažil v co největší míře využívat, jelikož to je nejrychlejší cesta k získání nutných informací k práci. Dalším nástrojem, který se ve firmě používá, je uzavřená skupina v produktu Skype. Tento nástroj jsme využili ke standardní psané komunikaci, ale také ke zprostředkování hovorů v případě večerního přesunu aplikace mezi dvěma cloudy.

Dalším důležitým nástrojem je open source aplikace Redmine. Tento nástroj slouží k elektronickému řízení projektů. V rámci této aplikace jsou zadávány úkoly jednotlivým vývojářům, které jsou rozděleny dle priorit. Každý úkol zde má svého zadavatele, svou prioritu, řešitele a datum, do kdy by měl být tento úkol vyřešen. Aplikace také umožňuje shlukovat související úkoly, tudíž je jasné, jaké úkoly se mají řešit přednostněji, jelikož na vyřešení jednoho úkolu může být závislé na řešení úkolu jiného. Tato aplikace slouží především pro komunikaci mezi projektovým managerem a vývojovým týmem, aby projektový manager byl schopen efektivně zadávat úkoly a také sledovat, v jaké fázi vývoje jsou již zadané úkoly.

Během vývoje používáme verzovací systém GIT. Z jeho funkcí jsem využívali především toho, že umožňuje více lidem pracovat nad stejnou částí aplikace, respektive upravovat stejnou část kódu. Dále tento nástroj efektivně udržuje historii kódu, větvení kódu na vývojovou a produkční větev. Což nám umožňuje vyvíjet nové funkce, ale zároveň opravovat chyby v kódu v produkčním nasazení. A také tento systém využíváme k nahrávání kódu na produkční servery.

Během praxe jsem úzce spolupracoval především s kolegy z vývojového týmu. Po zapracování jsem začal také přímo spolupracovat s projektovým managerem, který mi do té doby přiřazoval úkoly skrze jednoho z vývojářů. Po dosazení do pozice Frontend a JavaScript developer jsem začal také spolupracovat s firemním grafikem, který mi připravoval grafické návrhy aplikace a já je následně implementoval.

Hlavní náplní mé práce bylo navrhnout efektivní analytickou databázi, dále serverovou část, která slouží k dolování a zpracování dat a také klientskou část, která tato data zobrazuje v grafech. Během této práce jsem pomáhal s vývojem jiných částí aplikace, které bylo třeba vyvinout přednostněji.

3 Technologie

Při vývoji aplikace využíváme nepřehledné množství technologií souvisejících s vývojem webových aplikací v cloudu a pracování s big data. V této práci popisují především ty, které jsem využíval během odborné praxe.

3.1 Databázová část

MySQL MySQL je jeden z nejpoužívanějších relačních databázových systémů, který je založen na dialektu SQL. Data v této relační databázi jsou dělena do tabulek s definovanými sloupci, ve kterých je jeden záznam reprezentován jedním řádkem tabulky. Mezi tabulkami lze vytvářet logické vazby tak, aby byla zajištěna konzistence dat a aby mohla být data uložena v co nejvíce normalizované podobě.

Memcached Memcached je key-value úložiště v paměti pro menší část libovolných dat, která jsou získána z výsledků databázových dotazů, volání API, nebo renderování stránek. Slouží ke zvýšení rychlosti běhu programu tím, že časově náročné získávání dat je nahrazeno rychlým přístupem do paměti. Principem je, že po vybraných přístupech do databáze, API či renderování je výsledek uložen do Memcached a posléze jsou data přístupná již v paměti.

3.2 Serverová část

PHP PHP je skriptovací jazyk běžící na straně serveru. Slouží především k vytváření dynamických webových prezentací, či webových aplikací s vyžitím výstupních dat například ve formátu HTML. Pokud je PHP použit k vývoji API, nejpoužívanějším výstupem je formát XML nebo JSON.

Laravel Laravel je PHP framework pro vývoj webových aplikací vycházející z návrhového vzoru MVC. Framework je vyvíjen programátorem Taylorem Otwellm. Framework je poskytovaný zdarma jako open source projekt pod licencí MIT.

3.3 Klientská část

HTML Značkovací jazyk používaný pro tvorbu webových stránek, které jsou propojeny hypertextovými odkazy. Definuje sadu značek, atributu, chování a jejich topologii, které pak interpret (například webový prohlížeč) zobrazí v grafické podobě.

CSS CSS popisuje způsob zobrazení nebo chování elementů na webových stránkách napsaných v jazycích HTML, XHTML nebo XML. Hlavním smyslem je umožnit vývojářům oddělit vzhled dokumentu od jeho struktury a obsahu.

LESS LESS je jazyk pro tvorbu dynamického CSS, který je zkompilován do statického CSS. Samotný LESS se nevyužívá, slouží k přehlednějšímu a efektivnímu zápisu CSS stylů, které se z LESS generují.

3 TECHNOLOGIE

JavaScript JavaScript je skriptovací jazyk, který je spouštěn na straně klienta. Slouží především k dynamické reakci na uživatelskou akci.

jQuery JavaScriptová knihovna, která rozšiřuje stávající možnosti JavaScriptu. Jejím základním vlastností je snazší výběr HTML elementů a jejich modifikace. Usnadňuje také práci s událostmi, které jsou klíčové pro správu GUI pomocí JavaScriptu.

dxChart JavaScriptová knihovna, která slouží k vizualizaci dat do grafů ve formátu SVG. Doplňuje grafy o interakci s uživatelem formou znázornění vybraných částí grafů apod.

Ajax Označení pro vývoj webových stránek bez nutnosti jejich znovu načtení. Jedná se o kombinaci asynchronního JavaScriptu a XML. Slouží k vytvoření příjemnějšího uživatelského prostředí.

JSON JSON je způsob zápisu dat (datový formát), určený pro přenos dat, která mohou být organizována v polích nebo agregována do objektů. Jelikož je to pouze formát zápisu do řetězce, není vázán na jakýkoli programovací jazyk.

3.4 Architektura

MVC MVC je softwarová architektura, která dělí aplikaci do 3 logických vrstev jako datový model(Model), uživatelské rozhraní(View) a řídicí logiku(Controller) tak, že úprava některé z vrstev má co nejmenší dopad na funkčnost zbylých vrstev. Model obsahuje a zpracovává aplikační data a funkce, View zajišťuje prezentaci dat uživateli například za pomoci jazyka HTML a Controller zajišťuje správu interakce mezi uživatelem, modelem a pohledem.

4 Projekt MonkeyData

4.1 Popis projektu

Firma MonkeyData, s.r.o. vyvíjí aplikaci především pro e-shopy, které analyzují svá data o podnikání a chtějí data analyzovat efektněji. Základem projektu je to, že stahuje data z různých analytických nástrojů, ty pak zpracuje a zobrazuje v přehledných grafech, které například usnadňují analýzu toho jak se danému projektu daří nebo jak efektivně se využívají vynaložené finance na reklamu.

Aplikace slouží především provozovatelům e-shopu, kteří v aplikaci naleznou spoustu přehledných grafů na základě dat z jejich podnikání, a má sloužit jako podklad pro další vývoj daného e-shopu či projektu. Grafy jsou děleny do skupin podle jejich významu. Dominantní částí je přehled grafů, kde jsou zobrazeny nejdůležitější grafy a metriky, které by měl uživatel sledovat. Další grafy jsou děleny podle jejich zdroje dat a zaměření jako například návštěvnost webových stránek projektu na základě dat z Google Analytisc, či PPC, kde jsou zobrazeny data z internetové reklamy, pro kterou se v tuzemsku využívají především služby jako Sklik nebo Adwords. Navíc všechny tyto přehledy s grafy si může uživatel stáhnout v podobě PDF nebo jako balíček obrázků ve formátu PNG. Nebo si vybrané přehledy nechat posílat ve zvolených intervalech na email tak, aby mohl být neustále informován o svém podnikání.

Z technologického hlediska je aplikace rozdělena do několika částí, které by měli zajišťovat hladký a bezproblémový běh aplikace a její sto procentní dostupnost. Prvním faktorem je to, že celá aplikace je umístěna v cloudu od společnosti Google. Což umožňuje to, že pokud jeden ze serverů, na kterém aplikace jede vypadne ihned jej nahradí jiný, aniž by to uživatel jakýmkoliv způsobem poznal.

4.2 Členění projektu

4.2.1 Import dat

První částí aplikace je importní systém, který je pro samotné uživatele skryt, ale plní jednu z nejdůležitějších funkcí a to je neustále stahování dat ze zdrojů, které si uživatel v aplikaci napojil. Celá tato část běží na oddělených serverech tak, aby nijak nenarušila běh samotné aplikace. Importní část po stažení dat vše uloží tak, aby další části aplikace mohly tyto data efektivně zpracovat a upravit do finální podoby. Důležitou informací je to, že import dat probíhá nepřetržitě tak, aby uživatelé měli v aplikaci pořád co nejaktuálnější informace.

4.2.2 Kalkulace

Následuje hned za importní částí. Má za úkol očistit data o přebytečné informace a normalizovat data pomocí tzv. ETL nástrojů. Ty slouží k tomu, aby finální data byla v jednotné podobě uložena do databáze a mohly být nad nimi prováděny běžné databázové operace. Z důvodů obrovského množství dat, které není jen paměťově náročné, ale veškeré výpočty nad nimi jsou hlavně výpočetně náročné, jsme pro většinu práce s daty využili službu BigQuery firmy Google. Výsledná data jsou již ukládána do relační databáze a to

4 PROJEKT MONKEYDATA

konkrétně jako analytická databáze, kde jsou již efektivně uložena a připravena pro výběr dat do grafů.

4.2.3 Grafová část

Grafová část v aplikaci funguje v podstatě jako API, jen je zabezpečena tak, že k ní zle přistoupit pouze z aplikace samotné. Grafová část má za úkol výběr dat z databáze a všechna tato data připravit tak, aby byly připravené na vložení do grafů. V této části je kladen důraz na výkonnost celého běhu, jelikož jsou grafy načítány přímo na výstup k uživateli, tudíž musí být získání dat pro graf v řádech maximálně stovek milisekund.

Za součást grafové vrstvy se také počítá javascriptová část na straně klienta, která má za úkol vykreslit samotné grafy na výstup s využitím knihovny dxChart. Tako součást aplikace je také zodpovědná za zajištění komunikace se server a volání příslušných URL pro jednotlivé grafy. Také slouží k řízení obsahu grafu na základě uživatelských akcí.

4.2.4 Aplikační část

Aplikační část je hlavní částí aplikace, se kterou se uživatel setká. Slouží k zajištění komunikace s uživatelem. Skrze tuto část si uživatel vytváří projekty, ke kterým si napojí zdroje dat, které v daném projektu využívá a chce z nich vytvořit grafy. Napojení zdroje je nejčastěji koncipováno formou EasyClick nebo jen vyplnění nutných údajů k tomu, aby mohlo započít stahování dat. Také mu umožňuje sdílet své projekty s jinými uživateli, vytvářet si přehledné reporty a ty si nechat zasílat na email.

4.2.5 API

API slouží k získání dat aplikace zabezpečenou cestou tak, aby mohli být data využita například v mobilní aplikaci. Celé API je koncipováno formou REST API a přístup k němu je zabezpečen standardem OAuth2.

4.2.6 Mobilní aplikace

Mobilní aplikace slouží k tomu, aby uživatel měl skrze ní neustály přístup k datům bez nutnosti toho, aby byl nucen využít počítač. Aplikace slouží hlavně k zobrazení grafů a metrik tak jako ve webové aplikaci, nikoliv však k nastavení projektů či změnu napojení zdrojů, čímž je zajištěno co nejmenší využití prostředků mobilního telefonu nebo tabletu.

5 Zvolené řešení

Pro implementaci úkolů, které mi byli zadány jsem využil technologií a prostředků, které se ve firmě již používali například MySQL, PHP framework Laravel, HTML, CSS, JavaScript či knihovnu jQuery. Také bych rád zmínil jazyk LESS, které se na projektu nevyužíval, avšak já mám s tímto nástrojem dobré zkušenosti, tudíž po konzultaci s mým mentorem, jsem tento nástroj taktéž aktivně využil při řešení problémů. Při implementaci řešení jsem se snažil využívat návrhových vzorů a také zkušeností mých kolegů, kteří mi při návrhu asistovali.

5.1 Databázová část

Prvním z úkolů bylo navrhnout strukturu uložení dat do analytické databáze. Jako databázový systém jsme využili řešení MySQL. Při návrhu struktury jsem využil znalosti firemního analytika, který mě zasvětil do problému a já jsem následně na to uplatnil relační návrh. Vzhledem k objemu dat, která jsou skladována v databázi, jsem si rozdělil strukturu databáze na tabulky faktové (ty co obsahují pouze čistá data) a na tabulky katalogové (slovníky). A to z důvody možné duplicity dat, ale také proto to, abych byl schopen namapovat klíče, které stahujeme ze zdrojů na vlastní námi generované. Dále jsem data dělil podle uživatelů tak, aby jedna tabulka neobsahovala příliš mnoho záznamů, což by mohlo zásadně zpomalit dolování dat z databáze.

5.2 Serverová část

Pro řešení serverové části byl využit PHP framework Laravel pro jeho funkcionální základnu a jeho časovou efektivitu. Při řešení jsem z tohoto frameworku využil především routování pro formulaci URL dotazů. Dále jsem využil jeho implementaci přístupu k databázi, jelikož ji má velmi propracovanou a vyřešeny všechny problémy komunikace s databází. Celá serverová část je psaná objektové orientovanou metodou, pro její přehlednost a efektivnost závisující na rozsáhlosti projektu.

5.2.1 Laravel

Laravel framework, který je napsán v jazyce PHP a obsahuje spoustu řešení základních funkcionalit pro webovou aplikaci

- routování
- šablonovací systém Blade
- přístup k databázi
- cachování

5.2.1.1 Routování Routování je překlad mezi URL a požadavkem, který samotná aplikace zpracuje. Ze standardního překladu se dá snadno určit, jaký kontroler a jaká metoda bude danou URL zpracovávat a naopak z umístění jednotlivých metod se dají generovat URL adresy na danou metodu. Což se pak využívá k odkazu je jednotlivé části aplikace bez nutnosti zápisu statických URL, které se vlivem změny domény či struktury aplikace mohou měnit. Routování zajišťuje samostatná vrstva frameworku a její výhodou je to, že všechny generované URL jsou ve formátu tzv. friendly URL.

5.2.1.2 Šablonovací systém Blade Systém Blade slouží k vytváření šablon webové stránky. Šablony se vytvářejí z důvodu oddělení logické vrstvy(kontroler) od struktury a vzhledu stránky. Systém Blade vytváří náhradní syntaxi jazyka PHP, která se dá lépe začlenit do HTML struktury. Tyto šablony jsou následně parsovány do nativního PHP a jsou cachovány buďto na disk nebo do paměti.

5.2.1.3 Přístup k databázi Přístup k databázi je ve frameworku Laravel řešen skrze třídu QueryBuilder. Ta umožňuje pohodlné a dynamické skládání SQL dotazů, jejich správnou syntaxi a hlavně zabezpečení proti SQL injection. To je vyřešeno tím, že QueryBuilder všechny SQL dotazy formuluje jako parametrizované, čímž řeší problém SQL injection a také zefektivňuje dotazování, jelikož usnadňuje práci databázi s validací SQL dotazů.

5.2.1.4 Cachování Důležitou součástí frameworku je cachování. To zajišťuje to, že například dotazy, které již jednou zpracoval SQL server se uloží do paměti a při dalším volání stejného dotazu se již nezískávají z databáze nýbrž z paměti, která je mnohonásobně rychlejší než dotaz na databázi. Cachování lze využít také na uchování dat ze složitějších výpočtů nebo uložení neměnných struktur.

5.2.1.5 Základní cyklus Ve frameworku Laravel musí všechny žádosti vstoupit do public/index.php souboru. Ten dále volá soubor bootstrap/start.php, který vytvoří samotnou instanci aplikace a nastaví základní běhové prostředí. Dále je zde soubor framework/start.php, který má za úkol nakonfigurovat a načíst všechny služby. Dále jsou načteny aplikační soubory app/start, které obsahují rozšířená nastavení aplikace.

Následně jsou všechny dotazy odeslány do Route systému, který zajistí volání konkrétní metody konkrétního kontroleru. Ten požadavek zpracuje a vrátí výsledek formou HTTP odpovědi klientovi.

5.2.2 Implementace grafů

Základním konceptem je to, že každý konkrétní graf bude mít vlastní třídu, která bude zajišťovat získání všech dat pro daný graf. Aplikování nutných výpočtů specifických pro každý z grafů. Agregaci získaných dat dle časových období zvolených uživatelem a to zde dnů, týdnů, měsíců či roků. Dále zajištění konverze měn, do jednotné měny.

A transformaci dat do podoby, která je nutná pro využití v JavaScriptové knihovně na straně klienta.

Abych nemusel všechny mechanismy implementovat v jednotlivých grafem, využil jsem hierarchii dědičnosti, kdy jsem měl základní třídu, která obsahovala chování a společné a nezbytné parametry každého grafu. Základní třída také uchovává pořadí metod, které lze přetížít, aby byla umožněna libovolná specifičnost každého grafu. Nad základní třídou jsou vybudovány rozšiřující základní funkčnost pro jisté skupiny grafů s podobnou specifikací, aby nebylo nutné jejich chování specifikovat až na úrovni jednotlivých grafů. Z těchto rozšířených tříd již dědily konkrétní grafy, které většinou obsahovali jen základní informace o konkrétním grafu, které sloužily obecných implementacím k získání konkrétní podoby dat grafu.

5.3 Klientská část

Klientská část grafové vrstvy je napsaná v několika programovacích jazycích. Samotná struktura grafů byla psaná v jazyce HTML. Ta udržuje základní rozložení dat grafu a jejich umístění v prostoru pro jednotlivé grafy vymezeném.

Pro úpravu vzhledu jsem využil nástroj zvaný LESS. Ten umožňuje generovat minifikovanou strukturu CSS, která již přímo ovlivňuje HTML elementy na stránce. Toto jsem využil je grafické úpravě jednotlivých grafů. Zvýraznění důležitých částí grafu a skrytí méně podstatných údajů, které se však rozkryly po najetí kurzoru do určitého místa grafu.

Nejpodstatnější část klientské části grafové vrstvy je napsána v jazyce JavaScript s využitím knihoven jQuery a dxChart. Knihovnu jQuery jsem využil především z toho důvodu, že se v ní dá psát přehlednější kód a také se s její pomocí snáze upravují HTML elementy stránky oproti stejné funkcionalitě v čistém JavaScriptu. Tato knihovna navíc rozšiřuje základní funkcionality JavaScriptu, které jsem taktéž využil při implementaci.

Druhá z knihoven slouží k samotnému vykreslování grafů. Knihovna vykresluje grafy na základě dat a nastavení ve formátu SVG, který je velmi podobný jazyku HTML a dá se do něj také zakomponovat. Knihovnu jsem využíval pouze jako prostředek k znázornění dat v podobě grafu, všechny ostatní informace jako nadpisy grafu, sumarizační hodnoty a jednotky jsem řešil vlastním JavaScriptovým skriptem, který taktéž zajišťoval interakci s uživatelem. Data do grafů získával tento skript skrze technologii AJAX, která volala dotaz na pozadí stránky na server a umožnila to, že obsah grafů se mohl dynamicky měnit bez nutnosti aktualizace stránky. Také tento princip zajišťoval rychlý prvotní náběh stránky, jelikož časově náročné získávání dat pro grafy, probíhali zmíněnou AJAX technologií až po načtení stránky.

Skript také řeší vykreslování specifických grafů, které nejsou součástí knihovny dxChart pro jejich specifičnost. Také vytvářel doplňkové grafy k některým grafům, jelikož tuto funkcionalitu dxChart neobsahoval.

Druhý obdobný JavaScriptový skript obsluhoval vykreslování detailu grafu, kde je možnost si jeden typ dat nechat zobrazit v různých typech grafů. Což má umožnit uživateli rozsáhlejší pohled na danou metriku a také potencionálně snazší porozumění. Také tento skript zajišťuje rozšířené možnosti agregace dat, kdy si uživatel může sám zvolit,

5 ZVOLENÉ ŘEŠENÍ

jak se budou data agregovat, jelikož na přehledu grafů, se agregace vybírala programově na základě zvoleného období.

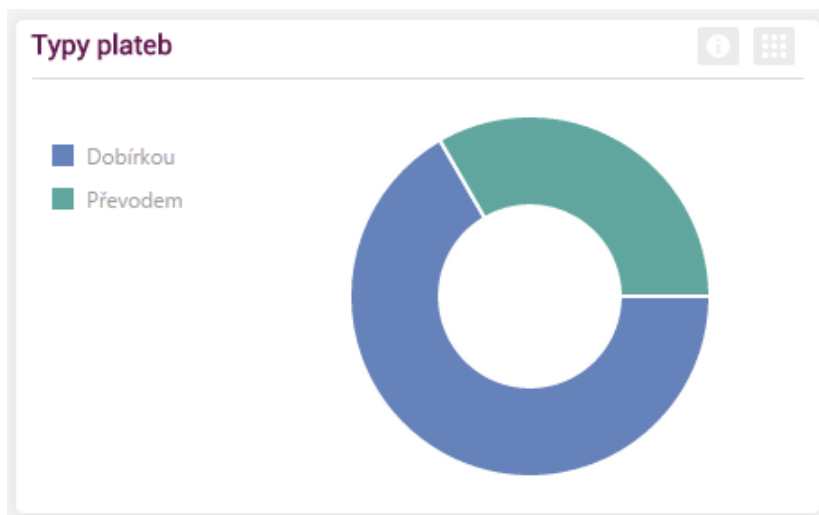
5.3.1 Typy grafů

Výpis grafů použitých v aplikaci a jejich specifika, která byla nutná zohlednit v jednotlivých částech grafové vrstvy.

5.3.1.1 Pie Koláčový graf (viz obr. 5.1), který se nejčastěji používá porovnání úzké množiny prvků. V kontextu e-shopu se tento graf například používá k porovnání vytižitosti jednotlivých možností platby, či počet nákupů v jednotlivých dnech týdne.

Specifikum tohoto typu grafu je jeho složitější dolování dat, jelikož je často nutné přistupovat do více než jedné tabulky v databázi.

Vykreslení tohoto grafu zajišťovala knihovna dxChart, vlastní script obsluhoval pouze umístění do grafu.



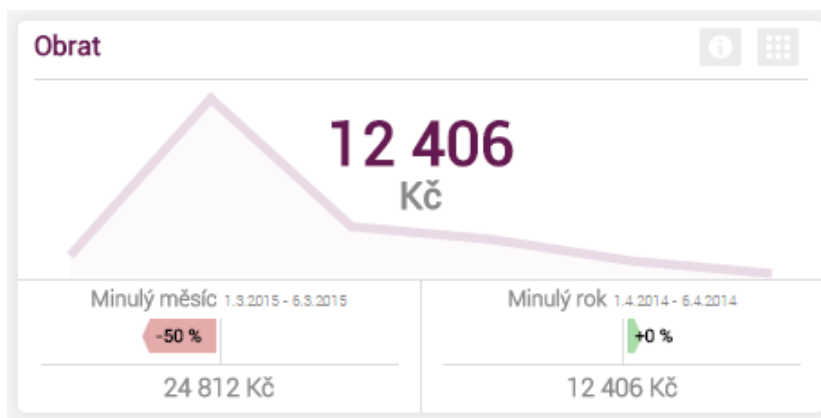
Obrázek 5.1: Typ grafu Pie

5.3.1.2 Spline Liniový graf (viz obr. 5.2), který se využívá k znázornění jedné či více hodnot v závislosti na čase. V kontextu e-shopu se tento graf například používá zobrazení obrátu či nákladů v závislosti na čase.

Specifikum tohoto typu grafu je jeho zpracování, jelikož se zde uplatňuje nejvíce modifikací jako konverze do jednotné měny a především agregace dat podle různých časových období a také doplnění období, pro která nejsou v databázi dostupná data. Další problematickým prvkem je sumarizační hodnota, která se většinou musela počítat zvlášť a také porovnání s jiným obdobím, kde se pro data muselo opakovaně přistoupit k databázi.

5 ZVOLENÉ ŘEŠENÍ

Vykreslení tohoto grafu zajišťovala knihovna dxChart, vlastní script obsluhoval umístění do grafu, výpis sumarizační hodnoty a také vykreslení doplňkového grafu pro porovnání období.

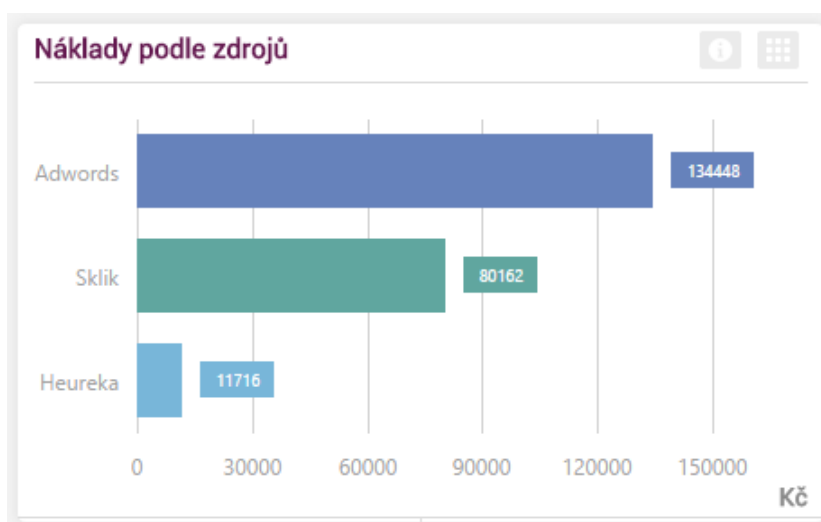


Obrázek 5.2: Typ grafu Spline

5.3.1.3 Bar Sloupcový graf (viz obr. 5.3), je obdobou grafu typu pie, který se nejčastěji používá porovnání menší množiny prvků. v kontextu e-shopu se tento graf například používá k porovnání kupní síly nejprodávanějších produktů.

Specifikum tohoto typu grafu je jeho složitější dolování dat, jelikož je často nutné přistupovat do více než jedné tabulky v databázi.

Vykreslení tohoto grafu zajišťovala knihovna dxChart, vlastní script obsluhoval pouze umístění do grafu.

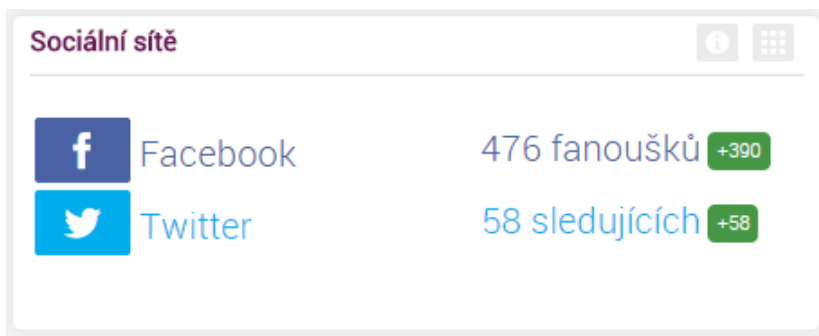


Obrázek 5.3: Typ grafu Bar

5 ZVOLENÉ ŘEŠENÍ

5.3.1.4 List Tabulkový graf (viz obr. 5.4), který se nejčastěji používá výčtu prvků určité množiny. V kontextu e-shopu se tento graf například používá vyjmenování neaktivnějších zákazníků, nebo stavu jejich stránek na sociálních sítích.

Vykreslení tohoto grafu zajišťoval pouze vlastní script, který generoval strukturu grafu ve formátu HTML a korektně umísťoval data do grafu.

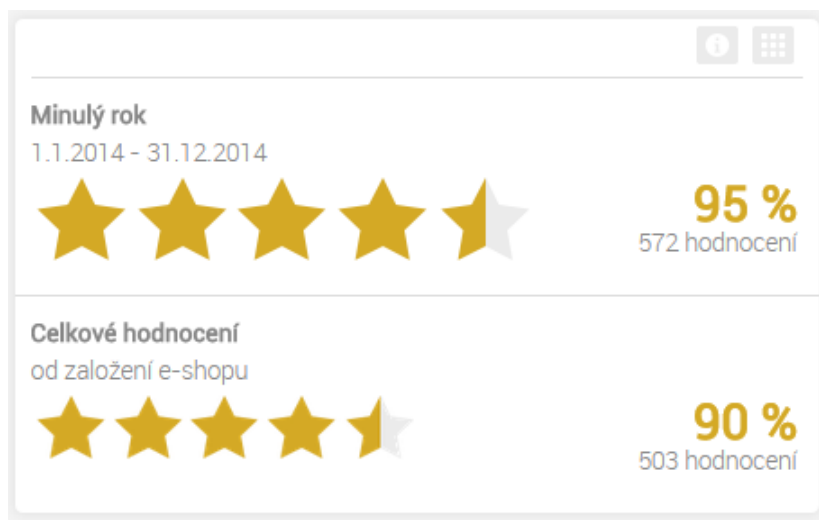


Obrázek 5.4: Typ grafu List

5.3.1.5 Star Hvězdicový či hodnotící graf (viz obr. 5.5), který se používá výhradně pro zobrazení dat získaných ze zdroje Heuréka. V kontextu e-shopu se tento graf například používá pro znázornění hodnocení e-shopu.

Specifikum tohoto typu grafu je poměrně snadné dolování dat, jelikož data pocházejí většinou pouze z jedné tabulky a nejsou na ně uplatněny žádné další modifikace.

Vykreslení tohoto grafu zajišťoval pouze vlastní script, který generoval strukturu grafu ve formátu HTML a korektně umísťoval data do grafu.



Obrázek 5.5: Typ grafu Star

5.4 Časový plán

V následující části se pokusím stručně popsat časovou náročnost jednotlivých úkonů během výkonu bakalářské praxe.

- 1 den** První den jsem se pouze seznámil s firemním prostředím
- 2 dny** Instalace počítače a všech nutných programů a přidělení SSH klíčů
- 2 dny** Seznámení se stávající verzí aplikace
- 1 den** Menší úkol za dohledu mentora
- 1 den** Specifikace zadání a seznámení s problematikou
- 3 dny** Návrh implementace zadaného úkolu včetně konzultace s mentorem
- 16 dní** Implementace zadaného úkolu
- 3 dny** Testování základní funkčnosti na pár vytipovaných grafech a ladění chyb
- 8 dní** Specifikace a implementace zbývajících grafů
- 4 dny** Testování funkčnosti všech grafů a ladění chyb, kompletní validace výstupů
- 5 dní** Tvorba nových šablon pro aplikace
- 4 dny** Výpomoc kolegům s úkoly, které se týkali jiných částí aplikace, které nespadali pod hlavní úkol bakalářské praxe
- 2 dny** Implementace nových grafů a jejich testování

6 Využité a nedostatečné znalosti

Před absolvováním odborné praxe jsem nepřikládal znalostem získaným během studia velkou váhu, jelikož na modelových příkladech v hodinách potenciál těchto znalostí nebyl využit. Ale jakmile jsem se dostal k řešení úkolů, kde řešení nebylo tak jednoznačné jako u ukázkových úloh, znalosti získané během studia se začaly být nezbytnými pro efektivní vyřešení problému. Tudíž až při odborné praxi jsem ocenil přínos informací, které jsem nabyl studiem na VŠB. A to nejen znalostí a zkušeností z výuky, ale také z kurzů, kterých jsem se mohl v rámci studia zúčastnit.

Při studiu na VŠB nám byl zprostředkován široký přehled informací z oblasti vývoje a návrhu softwarových řešení, získaných v předmětech jako softwarové inženýrství či vývoj informačních systémů, což jsem diskuzích o návrhu aplikace, kdy jsem na základě získaných znalostí mohl vnést do diskuze návrhy na řešení či využití některých z návrhových vzorů. Což bylo z mého pohledu velice přínosné pro můj další rozvoj, jelikož jsem se mohl aktivně zapojovat.

Jedním z úkolů bylo navrhnout strukturu databáze pro uložení dat a především k efektivnímu dolování dat z databáze. Zde jsem využil znalosti získané z předmětů zaměřených na návrh databází a v nemalé míře znalostí jak efektivně skládat SQL dotazy tak, aby jejich časová náročnost byla co nejmenší.

Na VŠB jsem přišel s jistou znalostí OOP, avšak tyto znalosti jsem si velice prohloubil v předmětech jako Algoritmy I, II nebo předmětu Vývoj Informačních Systémů. Takto získané znalosti jsem velice zúročil během řešení úkolů zadaných během praxe, protože aplikace je v OOP PHP.

Jelikož vývoj aplikace řeší 3 členný tým (včetně mě), projektový manažer a analytik, byla zde nezbytná komunikace a respektování firemních náležitostí, velkým přínosem mi byly předměty zaměřené na rozvoj měkkých dovedností a také absolvování kurzu Tieto IT Academy - Agile SW Development Game, kde se nám snažili nastítnit firemní procesy během vývoje. Kde jsme také měli možnost si vyzkoušet práci v týmu, práci pod tlakem, či jak prodat naši práci. A to vše ještě během studia.

Během praxe jsem měl možnost spolupracovat na návrhu serverové struktury v cloudu, což mě motivovalo k hlubšímu studiu této problematiky. Před nástupem na odbornou praxi jsem měl dobrou znalost jazyka PHP, avšak v průběhu praxe se naučil efektivně využívat dokumentování kódu tak, aby další vývoj aplikace byl snazší. Také jsem se zde naučil plně ovládat všechny možnosti verzovacího systému GIT, který ve firmě využíváme.

7 Závěr

Během řešení úkolu, které mi byly zadány v rámci odborné praxe jsem využil své znalosti, které jsem získal studiem na VŠB. Tyto znalosti mi velice usnadnili práci, jelikož jsem se jím nemusel učit až při samotném výkonu práce, což mi práci značně ulehčilo a urychlilo.

Byl jsem potěšen, když jsem do odborné diskuze k problému mohl vnést teoretická řešení problémů, se kterými jsem se měl možnost setkat v hodinách. Bylo pozoruhodné jak se informace, které se mi při výuce zdály bezcenné, ukázaly v praxi jako velice podstatné pro rozhodnutí, kterým směrem by se měl posunout návrh či vývoj určité části aplikace.

Největší zkušeností pro mě bylo vyzkoušet si vývoj aplikace dělený do několika etap, jako je vývoj, posléze test na testovacím serveru, kde se odhalily největší nedostatky, následně test na testovacím serveru, který však byl identický s produkčním serverem a následně nahrání na produkční servery. Všechny tyto operace byly časově náročné, avšak eliminovaly téměř všechny chyby, které potenciálně mohly nastat.

Celá aplikace je umístěna na serverech v cloudu, což pro mě byla novinka. Jelikož produkční server nebyl jeden, bylo nutné vzít v potaz tyto aspekty při vývoji aplikace, jelikož aplikace musela být psaná tak, aby byla maximálně škálovatelná. V této oblasti jsem se toho naučil nejvíce od svých kolegů, kteří se s touto problematikou již setkali a poskytli mi mnoho cenných rad.

Nejvíce si však cením zkušenosti práce v týmu, jelikož s ní se ve školském prostředí téměř nesetkáme. A také to, jak je nutné komunikovat se všemi členy týmu a vhodně zkombinovat znalosti všech, kteří se na projektu podílí.

Myslím si, že absolvovaná odborná praxe byla velice přínosná, hlavně pro můj profesní život, ale také studijní i osobní. Z mého pohledu jsem praxi absolvoval úspěšně a také tomu trochu napovídá to, že mi bylo nabídnuto, abych ve firmě nadále zůstal.

Literatura

SCHNEIDER, Robert D. *MySQL: oficiální průvodce tvorbou, správou a laděním databází*. 1. vyd. Praha: Grada, 2006, 372 s. ISBN 80-247-1516-3.

THE PHP GROUP. *PHP: Hypertext Preprocessor [online]*. © 2001-2015 [cit. 2015-05-03]. Dostupné z: <http://php.net/>

REDMINE. *Redmine [online]*. © 2006-2014 [cit. 2015-05-03]. Dostupné z: <http://www.redmine.org/>

TAYLOR OTWELL. *Laravel - The PHP Framework For Web Artisans [online]*. © 2012- [cit. 2015-05-03]. Dostupné z: <http://laravel.com/>

W3C. *World Wide Web Consortium (W3C) [online]*. © 2015 [cit. 2015-05-03]. Dostupné z: <http://www.w3.org/>